

# TER : Bras Robot pour jeu de plateau

Berriri Mehdi, Bounab Fayçal, Djema Sofiane, Doglio Arthur

June 2019

# Sommaire

I/	Introduction . . . . .	2
1/	Présentation du groupe . . . . .	2
2/	Présentation du sujet . . . . .	2
II/	État de l'art . . . . .	2
1/	Contexte . . . . .	2
2/	Technologies utilisées . . . . .	3
III/	Travail effectué . . . . .	5
1/	Études . . . . .	5
2/	Hardware . . . . .	10
3/	Software . . . . .	12
4/	Client et Serveur . . . . .	16
IV/	Gestion de projet . . . . .	17
1/	Organisation . . . . .	17
2/	Tâche . . . . .	18
3/	Problèmes rencontrés . . . . .	18
V/	Conclusion . . . . .	19
VI/	Perspectives et réflexions personnelles	20
1/	Perspectives . . . . .	20
2/	Réflexions personnelles . . . . .	20
VII/	Annexes . . . . .	21
1/	Journal de bord . . . . .	21

## **I/ Introduction**

Dans le cadre de notre de notre Master 1 Informatique nous avons souhaité prendre comme travail d'étude et de recherche le sujet qui porte le nom "Bras Robot pour jeu de plateau". Nous avons choisi ce sujet parce que nous pensons que les robots sont un excellent moyen pour expérimenter l'apprentissage par les essais et les erreurs.

### **1/ Présentation du groupe**

Notre groupe est composé de : Berriri Mehdi, Bounab Fayçal, Djema Sofiane, Doglio Arthur et nos intervenants sont : Mr Renavier Philippe et Mme Pelleau Marie .

### **2/ Présentation du sujet**

Le but de ce TER est de développer une librairie de commande du bras de robot Joy-It ROBOT02, ou améliorer une librairie existante et de mettre en oeuvre quelques mouvements dans un jeu réel, en fixant les emplacements ou en utilisant une librairie de reconnaissance pour les éléments d'un jeu.

## **II/ État de l'art**

### **1/ Contexte**

Notre bras est composé de six (6) cerveaux (moteurs) ; deux d'entre eux font des rotations (le premier cerveau qui sert de plate-

forme et le cinquième qui fait tourner la pince), un autre permet l'ouverture et la fermeture de la pince, et les derniers permettent la prise de la pièce d'échec. Ce bras évoluera dans un repère 3D ( $x, y, z$ ).

Nous avons commencé par le calibrage ; nous avons démonté chacun des cerveaux du robot afin de fixer ses positions initiales et son orientation. Pour cela nous avons calibré chacune des parties du bras c'est-à-dire faire en sorte que les différents moteurs ne s'entrechoquent pas entre eux . Nous avons aussi pris des multiples mesures pour trouver les différentes distance entre les différents éléments du robots (distance entre les moteurs) ainsi que les limites et les équivalences d'angles entre le réel et le code (disponible dans le document Angles sur github).

Nous sommes ensuite passé à la partie "mathématique" du TER c'est-à-dire calculer les différents angles de chacune des articulations afin d'atteindre l'objectif. Voici quelque représentation graphique du schéma général

Dans le code il y a une méthode de calcul des angles en prenant en compte des coordonnées ( $x, y, z$ ), une méthode de conversion (degrés en radian) et une méthode de déplacement des articulations.

## 2/ Technologies utilisées

Nous utilisons l'environnement de développement d'Arduino, le langage de programmation utilisé est le C++, compilé avec `avr-g++ 9`, et lié à la bibliothèque de développement Arduino, per-

mettant l'utilisation de la carte et de ses entrées/sorties. La mise en place de ce langage standard rend aisé le développement de programmes sur les plates-formes Arduino à toute personne maîtrisant le C ou le C++.



FIGURE 1 – Arduino IDE

Nous utilisons une carte Arduino qui est la base de la programmation avec microcontrôleur (un « mini-ordinateur » qui se présente sous la forme d'un circuit intégré). Il existe de nombreuses versions de cette carte. Actuellement nous avons une Arduino Uno wifi rev2



FIGURE 2 – Arduino Uno Wifi REV2

Les encadrants nous ont imposé le robot JOY-IT ROBOT02, ce bras robuste livré avec servomoteurs, pince et visserie de montage.

Le bras est équipé de 6 puissants servomoteurs de déplacement, 3 pince (rotation, inclinaison et ouverture/fermeture) et 3 pour le bras (coude, épaule et rotation de la base). Pour un coût de 176,40 euros



FIGURE 3 – Bras robotisé Joy-It ROBOT02

### III/ Travail effectué

#### 1/ Études

##### Arduino et Raspberry

##### Arduino Leonardo

La Leonardo est sortie après la carte Arduino Uno, et a été dotée d'un nouveau type de processeur, et d'une ergonomie revue. Elle a le même nombre de ports que la carte Arduino Uno (12 analogiques et 20 numériques dont 7 pwm). Son gros plus :

un port USB, qui vous permettra par exemple d'émuler un clavier ou une souris ! Cette carte sera très bien pour débiter un projet de domotique, comme la gestion de l'éclairage de sa maison ou la fermeture/ouverture des volets en fonction de la météo extérieure. Vous pouvez l'utiliser bien entendu sur des projets de niveau intermédiaire.

### Arduino Uno Wifi REV2

Cette Arduino possède (14 broches d'entrée/sortie numériques (dont 6 peuvent être utilisées comme sorties PWM), 6 entrées analogiques, un résonateur céramique de 16 MHz, une connexion USB, une prise d'alimentation, un en-tête ICSP et un bouton de réinitialisation. Elle comporte tout ce qui est nécessaire à la prise en charge du microcontrôleur. Il vous suffit de la connecter à un ordinateur à l'aide d'un cordon USB ou de l'alimenter avec un adaptateur c.a./c.c. ou une batterie. Le module Wi-Fi est un SoC autonome avec pile de protocole TCP/IP qui peut donner accès à votre réseau Wi-Fi. (Ou l'appareil peut agir en tant que point d'accès). Une caractéristique utile de Uno Wi-Fi est la prise en charge de la programmation OTA (over-the-air), que ce soit pour le transfert d'esquisses Arduino ou de firmware Wi-Fi. Il comprend un nouveau microprocesseur 8 bits à partir de Microchip Il est doté d'un IMU (unité de mesure d'inertie) Vous pouvez vous connecter à votre réseau Wi-Fi avec le module wifi intégré La connexion wifi est sécurisée par le nouvel accélérateur de puce cryptée ECC608)

### Raspberry Pi 3 Model B+



FIGURE 4 – Raspberry Pi 3

Le raspberry possède un processeur quad-core 64 bits de 1,4 GHz, LAN sans fil à double bande, Bluetooth 4.2/BLE, Ethernet plus rapide et prise en charge de l'alimentation sur Ethernet.

Pour conclure nous avons préféré rester avec l'arduino car nous avons déjà beaucoup travailler sur le code en C++. Sachant que sur raspberry c'est de la programmation en Python et aussi pour la raspberry nous avons besoin d'un écran ou bien de lancer une machine virtuel pour avoir accès à l'interface de la raspberry.

## **Bras Robotique**

Bras robotisé en kit FreaksArm EF08070





FIGURE 5 – Bras robotisé FreaksArm

Bras motorisé didactique commandé via une carte Freaduino compatible Arduino à assembler soi-même avec châssis en bois, servomoteurs permettant des mouvements sur 4 axes et 2 joysticks. Pour un coût de 75,90 euros.

Bras Dobot Basic



FIGURE 6 – Bras robotisé Dobot Basic

Le bras robotisé et modulaire Dobot Basic, destiné au milieu éducatif, permet de s'initier à la robotique, à la programmation et à la conception 3D. Ce bras accepte de nombreuses d'applications comme la réalisation de dessins, l'impression 3D et le déplacement d'objets. Les éléments permettant d'obtenir toutes ces fonctionnalités sont inclus dans le kit et sont facilement intégrables au bras (vis et cordon inclus). Le châssis est en aluminium anodisé et en plastique ABS assurant une bonne robustesse. Le conception de ce bras permet d'obtenir des opérations avec une précision de 0,2 mm grâce à ses 4 axes et ses différents moteurs. Il peut être contrôlé de plusieurs façons via un PC par cordon USB, par bluetooth (smartphone et tablette, application Android et iOS disponible) et peut être programmé en C++, C, Python et Java. Pour un coût de 1215 euros.

#### Bras robotique Braccio T050000



FIGURE 7 – Bras robotisé Braccio

Le bras motorisé et modulaire Braccio d'Arduino est livré en kit à assembler soi-même et se contrôle via une carte Arduino ou compatible (non incluse). Ce bras peut effectuer différents mouvements grâce à un ensemble de servomoteurs et dispose d'une pince pouvant saisir des objets jusqu'à 150 g. Le bras Braccio est polyvalent et sa modularité lui permet d'obtenir différentes configurations. Pour un coût de 238 euros.

Pour conclure, nous avons faits une étude sur les différents bras robotiques avec lesquels nous pourrions travailler. Avec le Frea-kArm est trop simple pour réaliser des mouvements dans notre projet. Le bras Dobot est un bras très bien pour pousser notre projet au maximum de ces capacités mais le prix est 10 fois plus cher que celui de notre bras actuel. Pour finir il y a le bras Braccio qui est 2 fois plusieurs chère, il est idéal aussi pour ce projet. Du coup c'est une question de prix.

## **2/ Hardware**

### **Assemblage et calibrage**

Nous avons eu à faire deux grandes étapes, l'assemblage du bras ainsi que son calibrage. Le calibrage représente tous les mouvements possibles de chaque articulation du bras. Celui-ci nous a été fourni assemblé mais nous l'avons démonté et remontré entièrement afin de calibrer les 6 servos comme suit :

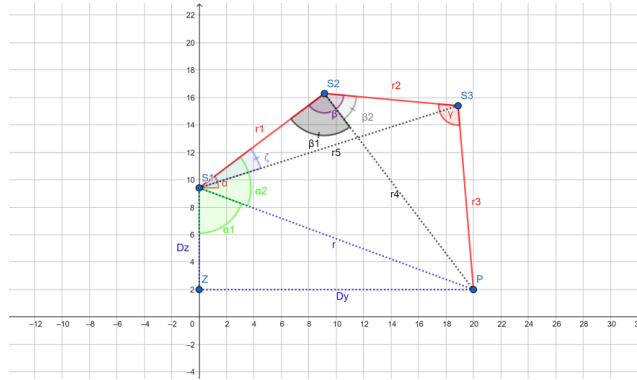


FIGURE 8 – Schéma 1

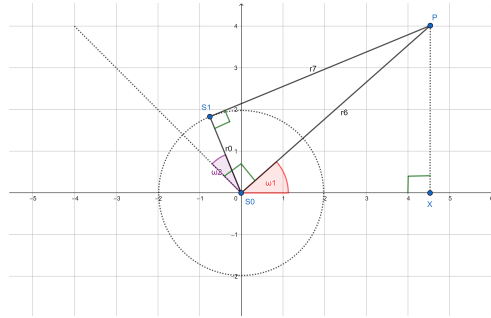


FIGURE 9 – Schéma 2

## Prise de mesures

Ensuite nous avons du prendre différentes mesures, les distances entre les servos, les angles de rotation grâce à des photos et un rapporteur virtuel.

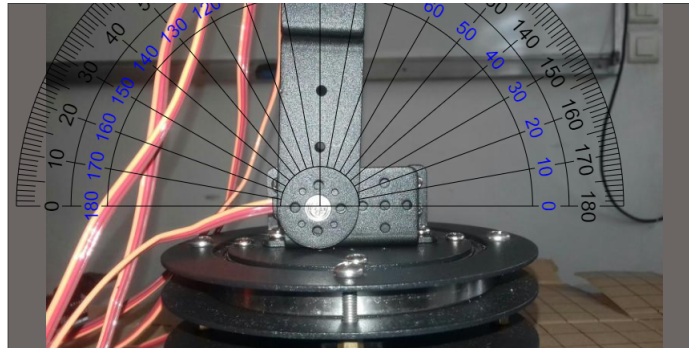


FIGURE 10 – Prise de mesure

### 3/ Software

#### Code

Voici les variables globales

```
float r0 = 2; // distance sur le plan(x,y) entre le Servo0 et le Servo1
float r1 = 10.4; // distance sur le plan(y,z) entre le Servo1 et le Servo2
float r2 = 9.8; // distance sur le plan(y,z) entre le Servo2 et le Servo3
float r3 = 14; // distance sur le plan(y,z) entre le Servo3 et la prise de la pince
float S1z = 9.4; // hauteur du Servo 1 en centimètre
```

- calculMoveDegre() : prend un angle en paramètre et calcul la valeur du déplacement du servo
- moveValeur() : déplace le Servo en position valeur passée en paramètre
- plateau() : Une classe qui représente un échiquier et un attribut n représentant le nombre de pièce adverses prises. Il y a aussi une méthode resetPlateau().

Nous allons parler des méthodes de calculs

- int cmTValeurServo5(float cm) : indique la position que doit

prendre le Servo5 (la pince) pour s'ouvrir selon la taille cm passée en paramètre.

- `int radianToDegre(float radian)` : converti des radian en degré.
- `float degreToRadian(float degre)` : converti des degrés en radian
- `calculAnglesArticulation(float x, float y, float z, int* tab)` : Elle prend en paramètre les coordonnées (x, y, z) de la pièce à prendre et un tableau où l'on sauvegarde les angles de rotation.

Tout d'abord, il faut orienter le bras vers sa destination. Pour cela, nous calculons la valeur de l'angle du Servo 0.

```
r6 = sqrt(x * x + y * y);  
r7 = sqrt(r6 * r6 - r0 * r0);  
omega1 = acos(x / r6);  
omega2 = (PI / 2) - acos(r0 / r6);  
omega = omega1 - (omega2 / 2);
```

Une fois cela fait, il faut déterminer les valeurs des angles des Servos formant l'articulation du bras. Pour cela, nous fixons l'angle du Servo 1 à 130 degré puis nous calculons les angles que doivent prendre les Servos 2 et 3 pour arriver à destination. Nous répétons cette opération pour un angle du Servo 1 décroissant de 5 en 5 jusqu'à 0.

```

Dy = abs(y - S1y);
Dz = abs(z - S1z);
r = sqrt(Dy * Dy + Dz * Dz);

if(z <= S1z){ alpha1 = asin(Dy / r);}
if(z > S1z){ alpha1 = (PI / 2) + asin(Dz / r);}
alpha = degreToRadian(Dalpha);
alpha2 = alpha + (PI / 2) - alpha1;

r4 = sqrt(r1 * r1 + r * r - 2 * r * r1 * cos(alpha2));
beta1 = acos((r1 * r1 + r4 * r4 - r * r) / (2 * r1 * r4));
beta2 = acos((r2 * r2 + r4 * r4 - r3 * r3) / (2 * r2 * r4));
beta = beta1 + beta2;
gamma = acos((r2 * r2 + r3 * r3 - r4 * r4) / (2 * r2 * r3));

```

Enfin, la combinaison d'angles que nous gardons est celle où le Servo 3 est le plus perpendiculaire à la coordonnées de destination.

```

r5 = sqrt(r1*r1 + r2*r2 - 2 * r1 * r2 * cos(beta));
zeta = acos(((r1*r1+r5*r5 - r2*r2)/(2*r1*r5)));
S3y = cos(alpha - zeta) * r5;
if(abs(S3y - y) < abs(distance - y) && angleS2 > 10 && angleS2 < 200 && angleS3 > 10
    distance = S3y;
    AS0 = angleS0;
    AS1 = angleS1;
    AS2 = angleS2;
    AS3 = angleS3;
}

```

Dans la classe main, nous avons le serveur et deux méthodes

- mouvementBras(float x, float y, float z, float ouverturePince, float fermeturePince, int typeMouvement) : Cette méthode permet de prendre (typeMouvement = 0) ou poser (typeMouvement != 0) une pièce(typeMouvement != 0).
- jouer(String instruction) : La méthode jouer permet d'effectuer un coup d'échec passé en paramètre (il faut qu'on explique nos choix pk le bras prends et pose ainsi)

## Tests

Nous avons testé la portée du bras. Il a une portée de 28 centimètres mais au delà de 23 centimètres, l'angle de la prise ou la pose de pièce n'est plus perpendiculaire ce qui fait que les pièces ne se posent pas correctement. Pour la précision, nous avons déplacé chaque pièce 10 fois entre 2 cases du plateau se trouvant dans la partie verte du schéma précédent et les pièces étaient posées dans leurs cases. Lorsque nous faisons ce même test avec au moins l'une des deux cases se trouvant dans la partie orange, la pièce n'est pas posée correctement et tombe. Nous avons ensuite mesuré la vitesse de rotation des Servos grâce à la fonction `millis()` d'arduino. Nous avons pris le temps avant le déplacement et après, puis nous avons fait la différence. Les résultats obtenus sont les suivants :

Voici le tableau des vitesses (secondes pour 100 degrés)



V	Servo 0	Servo 1	Servo 2	Servo 3	Servo 4	Servo 5	100 unité
12	3,25s	3,25s	3,25s	3,25s	2,6s	2,6s	1,3s
11	3s	3s	3s	3s	2,4s	2,4s	1,2s
10	2,75s	2,75s	2,75s	2,75s	2,2s	2,2s	1,1s
9	2,5s	2,5s	2,5s	2,5s	2s	2s	1s
8	2,25s	2,25s	2,25s	2,25s	1,8s	1,8s	0,9s
7	2s	2s	2s	2s	1,6s	1,6s	0,8s
6	1,75s	1,75s	1,75s	1,75s	1,4s	1,4s	0,7s
5	1,5s	1,5s	1,5s	1,5s	1,2s	1,2s	0,6s
4	1,25s	1,25s	1,25s	1,25s	1s	1s	0,5s
3	1s	1s	1s	1s	0,8s	0,8s	0,4s
2	0,75s	0,75s	0,75s	0,75s	0,6s	0,6s	0,3s
1	0,5s	0,5s	0,5s	0,5s	0,4s	0,4s	0,2s
0	0,25s	0,25s	0,25s	0,25s	0,2s	0,2s	0,1s

#### 4/ Client et Serveur

Le bras robot est contrôlé grâce à un client web. Au démarrage de l'Arduino celle-ci se connecte à un réseau et acquiert une IP. Lorsqu'un navigateur web se connecte à l'adresse de l'IP la carte Arduino lui renvoie un code HTML.

Avec le client on peut sélectionner une suite de coups et l'envoyer à l'Arduino. Lorsque le coup est envoyé le bras robot effectue sa tâche, pendant ce laps de temps le client s'actualise. La page finie de s'actualiser lorsque le coup est joué. A la fin de l'actualisation le plateau s'affiche à nouveau sur le client.

Voici l'interface client :

A8	B8	C8	D8	E8	F8	G8	H8
A7	B7	C7	D7	E7	F7	G7	H7
A6	B6	C6	D6	E6	F6	G6	H6
A5	B5	C5	D5	E5	F5	G5	H5
A4	B4	C4	D4	E4	F4	G4	H4
A3	B3	C3	D3	E3	F3	G3	H3
A2	B2	C2	D2	E2	F2	G2	H2
A1	B1	C1	D1	E1	F1	G1	H1

P

Envoyer

Effacer

## IV/ Gestion de projet

### 1/ Organisation

Niveau organisation on se voyait de façon hebdomadaire en fin de semaine aux campus des lucioles afin de travailler ensemble et discuter de l'avancement du Ter et si notre objectif hebdomadaire n'était pas atteint, nous continuions chez nous via l'application Discord. Nous organisions aussi des rendez-vous avec nos encadrants toute les trois semaines dans le but de leur montrer ce qu'on a fait et ce qu'on voudra faire pour les semaines à venir. De plus nous tenions un journal de bord chaque semaine qui est disponible sur notre git.

Dans le but de mieux s'organiser et d'avoir une trace écrite des

difficultés rencontrées nous avons mis en place un dépôt sur GitHub. Notre dépôt est composé de deux parties : une partie "Docs" qui contient différents fichiers (angles,etude,journal de bord) , et une partie "code" (les classe et les méthodes qui font bouger le bras).

Voici le lien du Github :

<https://github.com/faycalbounab/TER-Bras-Robot>

## 2/ Tâche

Voici la répartition du travail pour chaque membre du groupe (Arthur : A , Fayçal (F) , Mehdi (M) , Sofiane (S)) :

- Calibrage du bras (**A,M,S**)
- Prise des mesures (**M,S,F**)
- Implementation d'une interface client (**A,F**)
- Implementation d'un serveur (**A,S**)
- Découpage du code (classes et main) (**M,F**)

## 3/ Problèmes rencontrés

- Impossibilité de lancer un code dû à des problèmes de localisation du fichier source par rapport au dossier des librairies.
- Nous avons un petit décalage lors de la pose des pièces qui était causé par le fait que orientation du bras était calculé par rapport au coté gauche du l'équerre qui relie le moteur 1 au moteur 2
- La pince qui se desserre à chaque prise .Pour régler cela on a rajouté deux visses aux niveaux des deux crochets de la pince .

- L'arduino doit être brancher à un pc pour pouvoir l'alimenter suffisamment pour qu'une connexion avec un serveur-client ait lieu.

## V/ Conclusion

Dans le cadre de nos dernières années de licence en informatique nous avons eu l'occasion de faire plusieurs projets de programmation tous différents les uns des autres.

Pour notre première année de Master nous avons choisi de prendre un sujet qui combine et touche des domaines différents (Mathématique et Robotique). C'est le côté découverte du qui nous attirait dans ce projet, nous voulions des nouveaux outils qui pourraient enrichir notre expérience dans le domaine de l'informatique.

Lorsque nous avons entamé ce projet nous avions un peu mal estimé la charge de travail mais cela nous a aidé à mieux nous organiser et travailler de façon plus efficace . Grâce aux notions que nous avons acquises ces dernières années nous avons trouvé que la partie software était assez abordable et heureusement que nous avions une bonne cohésion d'équipe parce que tout le côté interaction hardware-software était un peu complexe (bien calibrer le bras, faire des mouvements précis au bras...).

Pour conclure, le sujet que nous avons choisi nous a vraiment permis de mettre en application nos connaissances en informatique tout en obtenant des nouvelles notions, dans le futur ce sera plus facile pour nous de refaire des projets de ce genre

## VI/ Perspectives et réflexions personnelles

### 1/ Perspectives

Pour aller plus loin dans ce projet, nous pourrions utiliser des capteurs afin de développer une IA pour pouvoir jouer sans à avoir à donner d'instructions aux bras dans une partie d'échecs avec une autre IA ou bien une personne sachant jouer.

### 2/ Réflexions personnelles

- **Berriri Mehdi** : J'ai trouvé le projet passionnant autant du point de vue des découvertes de la robotique que de la programmation Arduino. J'ai aimé développer en Arduino et découvrir cet outil. Ce projet m'a permis d'appliquer mes connaissances en mathématiques et informatique à un projet qui me tient à coeur.
- **Bounab Fayçal** : Ce Ter m'a personnellement énormément apporté, et grâce au travail que j'ai effectué sur le hardware et le software. Au début ce n'était pas facile mais à force de redoubler d'efforts je ressors plus expérimenté et plus aptes à faire des projet sur des longues durées
- **Djema Sofiane** : J'ai trouvé ce projet très passionnant dans le sens où il m'a fait découvrir une facette de l'informatique qui ne m'intéressait pas à première vue et je parle bien sûre de la robotique. Le fait d'avoir un résultat visuel de ce qu'on a codé est une expérience assez satisfaisante et si je devais retravailler sur un projet de ce type, ce serait un grand plaisir de le faire.

- **Doglio Arthur** : Je suis du même avis que mes collègues en ce qui concerne la découverte de la robotique. Ce projet était très intéressant par son aspect multidisciplinaire. Avec ce dernier, j'ai pu revoir mes formules de mathématiques et je les ai réutilisées. J'ai aussi appris beaucoup de choses qui m'ont permis de mieux comprendre les interactions hardware-software.

## VII/ Annexes

### 1/ Journal de bord

#### **Semaine du 04/03/2019**

Nous avons récupéré le bras robotique auprès de nos encadrants, nous allons commencer par calibrer le bras avant de réfléchir au jeux que nous allons étudier. Nous pensons déjà travailler sur le jeu de société Takenoko et le jeu d'échec. Dans ce TER nous ferons une étude sur les différents bras robotiques que nous aurions pu avoir et voir aussi la différence entre un arduino et une raspberry.

#### **Semaine du 11/03/2019**

Cette semaine nous avons démonté et remonté tous les moteurs du bras robotique afin de les calibrer. Après avoir fini l'assemblage nous avons remarqué que ses mouvements étaient trop brusques de ce fait nous avons diminué la vitesse de ses mouvements. Pour la semaine prochaine on essayera de lui faire prendre un objet quelconque pour avoir une vision plus précise des possibilités offertes par le robot.

### **Semaine du 19/03/2019**

Cette semaine nous avons réussi à faire attraper et reposer une pièce du jeu d'échecs sans la faire retomber. Pour la semaine prochaine on essaiera de faire des calculs sur les angles pour faire en sorte que les différentes parties (surtout les parties gérées par le moteur 2,3,4) du robot ne se gênent pas lors du mouvement.

### **Semaine du 25/03/2019**

Cette semaine nous avons préparé un quadrillage pour positionner en (x,y) le bras dans l'espace. Puis, nous avons commencé à calculer les angles de déplacement des moteurs et leur position initial dans l'espace. Pour la semaine prochaine nous continuerons sur les calculs afin de les implémenter.

### **Réunion avec Monsieur Rennevier**

Nous avons fait une démonstration du bras robotique, il a ainsi fait une prise d'une pièce d'échec. Ensuite nous avons vu avec Monsieur Rennevier s'il y a la possibilité d'intégrer le wifi à notre carte arduino. Nous allons faire une recherche sur la carte wifi à commander et sur les différences entre les raspberrys et notre carte arduino.

### **Semaine du 01/04/2019**

Cette semaine nous avons fini les calculs des angles(voir fichier Angles), nous avons commencé un système de contrainte(formules) pour avoir le champ d'action du bras robotique. Pour le découpage du code, nous avons fait une classe servo pour avoir des objets qui correspondent à chacun des servo.

### **Semaine du 08/04/2019**

Cette semaine nous avons fait un bilan avec Monsieur Malapert, nous avons testé des formules qui permettent de calculer les angles afin d'effectuer un mouvement avec le bras. Nous intégrerons ces formules au code par la suite. Nous allons aussi faire une étude en parallèle sur la possibilité de passer sur Raspberry.

### **Semaine du 15/04/2019**

Cette semaine nous avons mis en ligne les calculs via un fichier HTML, qui renvoie vers une page où l'on peut préciser les coordonnées dans l'espace(x,y). Il y a aussi en ligne le début de l'étude des bras robotiques. Nous avons recalibré le Servo4. Enfin nous avons intégré une fonction de calcul des angles de rotations des 4 premiers Servo.

### **Semaine du 22/04/2019**

Cette semaine nous avons fait deux fonctions qui renvoient les angles de rotation des 4 premiers Servos sous forme d'un tableau. Nous lisons les coordonnées x, y et z en entrée standard que nous utilisons pour effectuer un mouvement.

### **Semaine du 29/04/2019**

Cette semaine nous avons implémenté la prise et la pose d'une pièce (on rentre des positions pour la position de la pronation, le bras se déplace vers la position indiquée, il prend la pièce et se met en position initiale puis attend les nouvelles coordonnées qui indiquent la position où il doit déposer la pièce). Une diminution des mouvements incontrôlables lors du 1er mouvement. Et un moyen de contrôler la vitesse des servos.

### **Semaine du 06/05/2019**



Cette semaine nous avons implémenté deux approches permettant de calculer les mouvements que doivent effectuer les Servos du bras pour se déplacer vers une coordonnée  $(x,y,z)$ .

### **Réunion avec Madame Pelleau**

Nous avons présenté les mouvements de prise et pose d'un fou, cependant il y avait encore des imprecisions causées par une méthode de calcul des déplacements qui n'était pas aboutie. Nous avons discuté d'une possible augmentation de la portée du bras grâce à de plus grandes équerres. Nous avons aussi dit que nous avons l'intention de travailler sur le contrôle à distance du bras grâce à une raspberry et l'arduino que Mr Renevier nous a déposé. Mme Pelleau nous a aussi laissé un module wifi pour arduino.

### **Semaine du 13/05/2019**

Nous avons recalibré les Servos 2 et 3. Nous commençons le travail sur l'arduino wifi et nous avons essayé de faire en sorte de bouger plusieurs servos en même temps mais cela n'est pas possible car le temps de calcul nécessaire est trop important.

### **Réunion avec Mr Rennevier**

Nous avons présenté les mouvements de prise et pose d'un fou. Nous avons parlé du jeu d'échec, nous allons faire jouer le bras sur un plateau de taille 5x5. Nous essayons d'activer plusieurs servos en même temps et faire fonctionner le wifi. Mr Rennevier nous a conseillé de tester les limites du bras ( précision et portée ), quelles informations intégrer dans le rapport (calibrage du bras ) et comment réaliser une vidéo pour la présentation (durée de la

vidéo et présenter une partie en peu de coup).

### **Semaine du 20/05/2019**

Cette semaine, nous avons repris des mesures plus précises des rotations des Servos (en prenant des photos et avec un rapporteur virtuel). Nous avons testé l'arduino wifi qui fait fonctionner le bras correctement et nous avons commencé à faire un client pour celle-ci. Nous ferons une pause la semaine prochaine.

### **Semaine du 03/06/2019**

Cette semaine, nous avons commencé le rapport et nous avons commit une première version du rapport, il n'est pas complet il manque encore des informations que nous aurons d'ici à la semaine du 10/06. Pour le moment c'est surtout pour avoir une idée très large de comment il sera. Nous avons aussi fini les calculs de notre bras pour la précision de celui-ci grâce à une méthode que nous avons faite et nous sommes toujours entrain de bosser en parallèle le Wi-fi.

### **Réunion avec Mr Rennevier**

Nous avons présenté notre client et notre serveur qui est pour le moment opérationnelle il nous reste à l'intégrer à notre code afin de tester, nous avons présenté une première version du rapport et pour finir nous avons montré l'amélioration des mouvements du Bras.